|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №6**

Выполнил студент группы ИНБО-02-20 Абдусалямова С. К.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

Упражнение 1.

Написать тестовый класс, который создает массив класса Student и сортирует массив iDNumber и вставками.

Упражнение 2.

Напишите класс SortingStudentsByGPA который реализует интерфейс Comparator таким образом, чтобы сортировать список студентов по их итоговым баллам в порядке убывания с использованием алгоритма быстрой сортировки.

Упражнение 3.

Напишите программу, которая объединяет два списка данных о студентах в один отсортированный списках с использованием алгоритма сортировки слиянием.

## **Ход Работы**

Код программы:

//StudentTest.java

package com.company;  
  
  
import java.util.ArrayList;  
  
public class StudentTest {  
 public static void main(String[] args) {  
 Student[] students = new Student[]{  
 new Student(133, 38),  
 new Student(127, 45),  
 new Student(159, 99),  
 new Student(118, 74)  
 };  
 System.*out*.println("Неотсортированный массив:");  
 for (Student s : students) {  
 System.*out*.println(s);  
 }  
 System.*out*.println();  
  
 System.*out*.println("отсортированный по ID массив:");// Вставки по id  
 Student temp;  
 for (int i = 1; i < students.length; i++) {  
 Student current = students[i];  
 int j = i-1;  
 for(; j >= 0 && current.compareTo(students[j]) < 0; j--) {  
 students[j+1] = students[j];  
 }  
 students[j+1] = current;  
 }  
 for (Student s : students) {  
 System.*out*.println(s);  
 }  
 System.*out*.println();  
  
 *qSort*(students, students.length-1, 0);  
 System.*out*.println("отсортированный по GPA массив:");// Быстрая по GPA  
 for (Student s : students) {  
 System.*out*.println(s);  
 }  
 System.*out*.println();  
  
 Student[] students2 = new Student[]{  
 new Student(185, 41),  
 new Student(164, 26),  
 new Student(113, 32),  
 new Student(119, 83)  
 };  
 Student[] allStudents = new Student[students.length + students2.length];  
 System.*arraycopy*(students, 0, allStudents,0,students.length);  
 System.*arraycopy*(students2, 0, allStudents,students.length, students2.length);  
 *mergeSort*(allStudents, allStudents.length);  
 System.*out*.println("объединённый отсортированный по ID массив:");// merge sort  
 for (Student s : allStudents) {  
 System.*out*.println(s);  
 }  
 }  
  
 public static void mergeSort(Student[] a, int n) {  
 if (n < 2) {  
 return;  
 }  
 int mid = n / 2;  
 Student[] l = new Student[mid];  
 Student[] r = new Student[n - mid];  
  
 System.*arraycopy*(a, 0, l, 0, mid);  
 System.*arraycopy*(a, mid, r, 0, n - mid);  
 *mergeSort*(l, mid);  
 *mergeSort*(r, n - mid);  
  
 *merge*(a, l, r, mid, n - mid);  
 }  
 public static void merge(  
 Student[] a, Student[] l, Student[] r, int left, int right) {  
  
 int i = 0, j = 0, k = 0;  
 while (i < left && j < right) {  
 if (l[i].compareTo(r[j]) <= 0) {  
 a[k++] = l[i++];  
 }  
 else {  
 a[k++] = r[j++];  
 }  
 }  
 while (i < left) {  
 a[k++] = l[i++];  
 }  
 while (j < right) {  
 a[k++] = r[j++];  
 }  
 }  
  
  
 private static StudentComp *comp* = new StudentComp();  
  
 public static void qSort(Object[] array, int high, int low){  
 if(array == null || array.length == 0) return;  
 if(high <= low) return;  
  
 Object middle = array[(low + high)/2];  
 ArrayList<Object> left = new ArrayList<>();  
 ArrayList<Object> right = new ArrayList<>();  
 ArrayList<Object> eq = new ArrayList<>();  
 for(int i = low; i <= high; i++){  
 if(*comp*.compare(array[i], middle) > 0){  
 left.add(array[i]);  
 }  
 else if (*comp*.compare(array[i], middle) < 0)  
 right.add(array[i]);  
 else eq.add(array[i]);  
 }  
 Object[] leftArr;  
 Object[] rightArr;  
 if(left.size()>0) {  
 leftArr = left.toArray();  
 *qSort*(leftArr, left.size() - 1, 0);  
 System.*arraycopy*(leftArr, 0, array, low, left.size());  
 }  
 System.*arraycopy*(eq.toArray(), 0, array, low+left.size(), eq.size());  
  
 if(right.size() > 0) {  
 rightArr = right.toArray();  
 *qSort*(rightArr, right.size() - 1, 0);  
 System.*arraycopy*(rightArr, 0, array, low+left.size() + eq.size(), right.size());  
  
 }  
  
 }  
}

//Student.java

package com.company;  
  
public class Student implements Comparable {  
  
 private int idNum;  
 private int GPA;  
  
 public Student(int idNum, int GPA) {  
 this.idNum = idNum;  
 this.GPA = GPA;  
 }  
  
 public String toString() {  
 return "Student{" +  
 "idNum=" + idNum +  
 ", GPA=" + GPA +  
 '}';  
 }  
  
 public int getIdNum() {  
 return idNum;  
 }  
  
 public void setIdNum(int idNum) {  
 this.idNum = idNum;  
 }  
  
 public int getGPA() {  
 return GPA;  
 }  
  
 public void setGPA(int GPA) {  
 this.GPA = GPA;  
 }  
  
 @Override  
 public int compareTo(Object o) {  
 if(!(o instanceof Student))  
 throw new IllegalArgumentException("Объект не является студентом!");  
 // < 0 -> o, 0 -> ==, >0 -> this;  
 return this.idNum - ((Student) o).idNum;  
 }

}

//StudentComp.java

package com.company;  
  
import java.util.Comparator;  
  
public class StudentComp implements Comparator {  
 @Override  
 public int compare(Object o1, Object o2) {  
 if(!(o1 instanceof Student && o2 instanceof Student))  
 throw new IllegalArgumentException("Объекты не являются студентами!");  
  
 return ((Student) o1).getGPA() - ((Student) o2).getGPA();  
 }

## **Вывод**

В ходе практической работе, я научилась работать с интерфейсом Comparator и алгоритмами сортировки на языке Java.